

Esercizio 1. Siano dati i seguenti numeri binari in rappresentazione con virgola mobile (1 bit di segno, 6 di mantissa e 3 di esponente): $\langle 0,100100,001 \rangle$ e $\langle 0,110000,111 \rangle$.

a) Se ne effettui il prodotto, eventualmente normalizzando il risultato (3 punti):

Procedimento:

Per poter interpretare i due numeri bisogna convertire l'esponente, che è espresso in complemento a 2. Nel primo caso il bit più significativo è 0, quindi l'esponente è positivo e facilmente si ottiene 1 convertendo in decimale. Nel secondo caso il bit più significativo è 1, quindi l'esponente è negativo e per convertirlo sottraggo 1 (ottenendo 110), lo complemento bit a bit (ottenendo 001) che considerando il segno corrisponde a -1. L'esponente del risultato sarà la somma dei due esponenti, quindi 0. Si effettua quindi il prodotto in colonna delle mantisse 0,1001 e 0,11:

0,1001
0,11
01001
01001
00000
0011011

Per normalizzarlo si deve spostare la virgola di una posizione a destra, ottenendo 0,11011 e portando l'esponente a -1.

Risultato: $\langle 0,11011,111 \rangle$

b) Si converta il risultato ottenuto in base 10 (1 punto):

Procedimento:

Il risultato al punto precedente può essere scritto come $0,11011_{(2)}$.

Effettuiamo il procedimento di conversione:

$$2^{-2} + 2^{-3} + 2^{-5} + 2^{-6} = \mathbf{0,421875}$$

c) Si converta il numero così ottenuto in base 5, fermandosi alla 4^a cifra dopo la virgola (2 punti):

Procedimento:

Operazione	Parte Intera	Parte Decimale
$5 \times 0,421875 = 2,109375$	2	0,109375
$5 \times 0,109375 = 0,546875$	0	0,546875
$5 \times 0,546875 = 2,734375$	2	0,734375
$5 \times 0,734375 = 3,671875$	3	0,671875

Le prime 4 cifre della parte decimale sono quindi $0,2023_{(5)}$.

Risultato: $\mathbf{0,2023}$

Esercizio 2. Si consideri la seguente stringa binaria: 1001.

a) Se ne calcoli il bit di parità pari (1 punto): Risposta: $\mathbf{0}$

b) Si scriva la stringa come una matrice 2x2 e se ne calcolino i bit di parità pari longitudinale e trasversale (1 punto):

1	0	1
0	1	1
1	1	

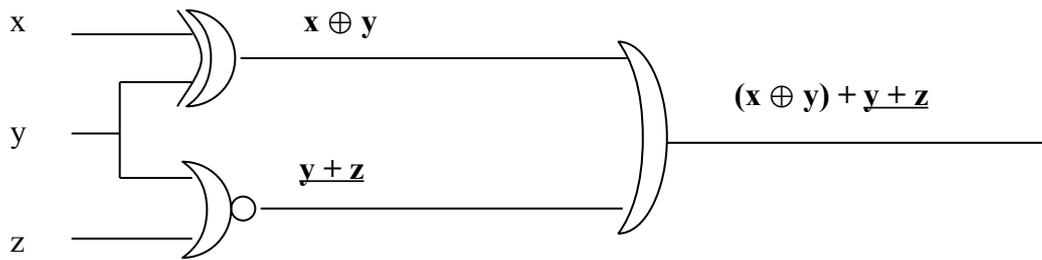
c) Si scriva la parola di codice di Hamming associata alla stringa di bit data (2 punti):

Procedimento:

La parola sarà XX1X001, il primo bit controlla la parità dei bit 1, 3, 5, 7, il secondo quella dei bit 2, 3, 6, 7, il quarto quella dei bit 4, 5, 6, 7.

Risultato: **0011001**

Esercizio 3. Si consideri il seguente circuito combinatorio:



a) Si scriva sul disegno, in corrispondenza di ogni linea di uscita di ogni porta, l'espressione booleana calcolata dalla porta (1 punto)

b) Si calcoli la forma canonica SOP (o forma canonica disgiuntiva) associata all'espressione finale (5 punti)

Procedimento:

$$\begin{aligned}
 (x \oplus y) + \underline{y} + \underline{z} &= && \text{(dalla definizione dello XOR e de morgan)} \\
 = x \underline{y} + \underline{x} y + \underline{y} \underline{z} &= && \text{(applicando de morgan)} \\
 = x \underline{y} (z + \underline{z}) + \underline{x} y (z + \underline{z}) + \underline{y} \underline{z} (x + \underline{x}) &= && \text{(propr. Distributiva e idempotenza)} \\
 = x \underline{y} z + x \underline{y} \underline{z} + \underline{x} y z + \underline{x} y \underline{z} + \underline{x} \underline{y} \underline{z} &= &&
 \end{aligned}$$

c) Si scriva la tavola di verità della funzione booleana associata (2 punti)

x	y	z	f(x, y, z)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Esercizio 4. Si vuole realizzare un circuito combinatorio associato alla funzione booleana di 4 variabili binarie di ingresso e due uscite binarie che restituisce in output la codifica binaria di quante linee in ingresso valgono 0. Si assuma che le variabili di ingresso non siano mai tutte contemporaneamente a 0.

Esempio: se l'input è 0101, l'output sarà 10 (la codifica binaria del numero 2)

a) Si scriva la tavola di verità della funzione (2 punti)

X_0	X_1	X_2	X_3	Y_1	Y_0
0	0	0	0	X	X
0	0	0	1	1	1
0	0	1	0	1	1
0	0	1	1	1	0
0	1	0	0	1	1
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	1	1
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	0	0

b) Si calcoli la forma normale SOP minimale per il bit più significativo della funzione (2 punti)

Procedimento:

		X_0		X_1	
		00	01	11	10
X_2	00	X	1	1	1
	01	1	1	0	1
	11	1	0	0	0
	10	1	1	0	1

$$Y_1 = \overline{X_0}\overline{X_1} + \overline{X_2}X_3 + \overline{X_1}X_3 + \overline{X_0}X_3 + \overline{X_1}X_2 + \overline{X_0}X_2$$

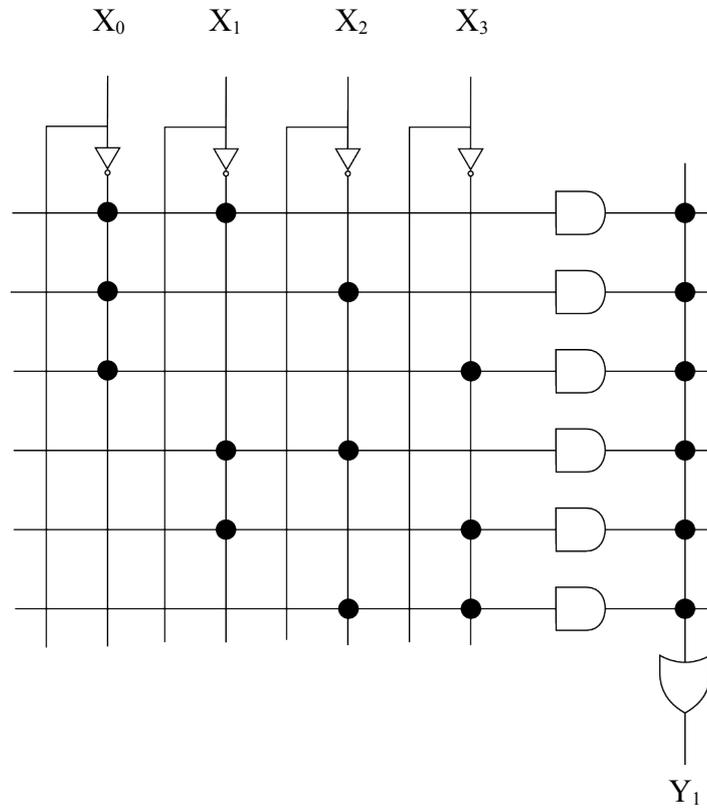
c) Si semplifichi l'espressione ottenuta usando algebra di Boole e altre porte logiche (2 punti)

Procedimento:

$$\begin{aligned} \overline{X_0}\overline{X_1} + \overline{X_2}X_3 + \overline{X_1}X_3 + \overline{X_0}X_3 + \overline{X_1}X_2 + \overline{X_0}X_2 &= \\ = \overline{X_0}(\overline{X_1} + X_3) + \overline{X_2}(\overline{X_1} + X_3) + \overline{X_1}X_3 + \overline{X_0}X_2 &= \\ = (\overline{X_0} + \overline{X_2})(\overline{X_1} + X_3) + \overline{X_1}X_3 + \overline{X_0}X_2 & \end{aligned}$$

$$\begin{aligned}
 &= (\underline{X_0 X_2}) (\underline{X_1 X_3}) + (\underline{X_1 + X_3}) + (\underline{X_0 + X_2}) \\
 &= (\underline{X_0 X_2 + X_1 X_3}) + (\underline{X_1 + X_3}) (\underline{X_0 + X_2}) \\
 &= (\underline{X_0 X_2 + X_1 X_3}) ((\underline{X_1 + X_3}) (\underline{X_0 + X_2}))
 \end{aligned}$$

d) Si realizzi tramite PLA l'espressione ottenuta al punto (b) (2 punti)
 (N.B.: è necessario visualizzare la matrice di AND e di OR)

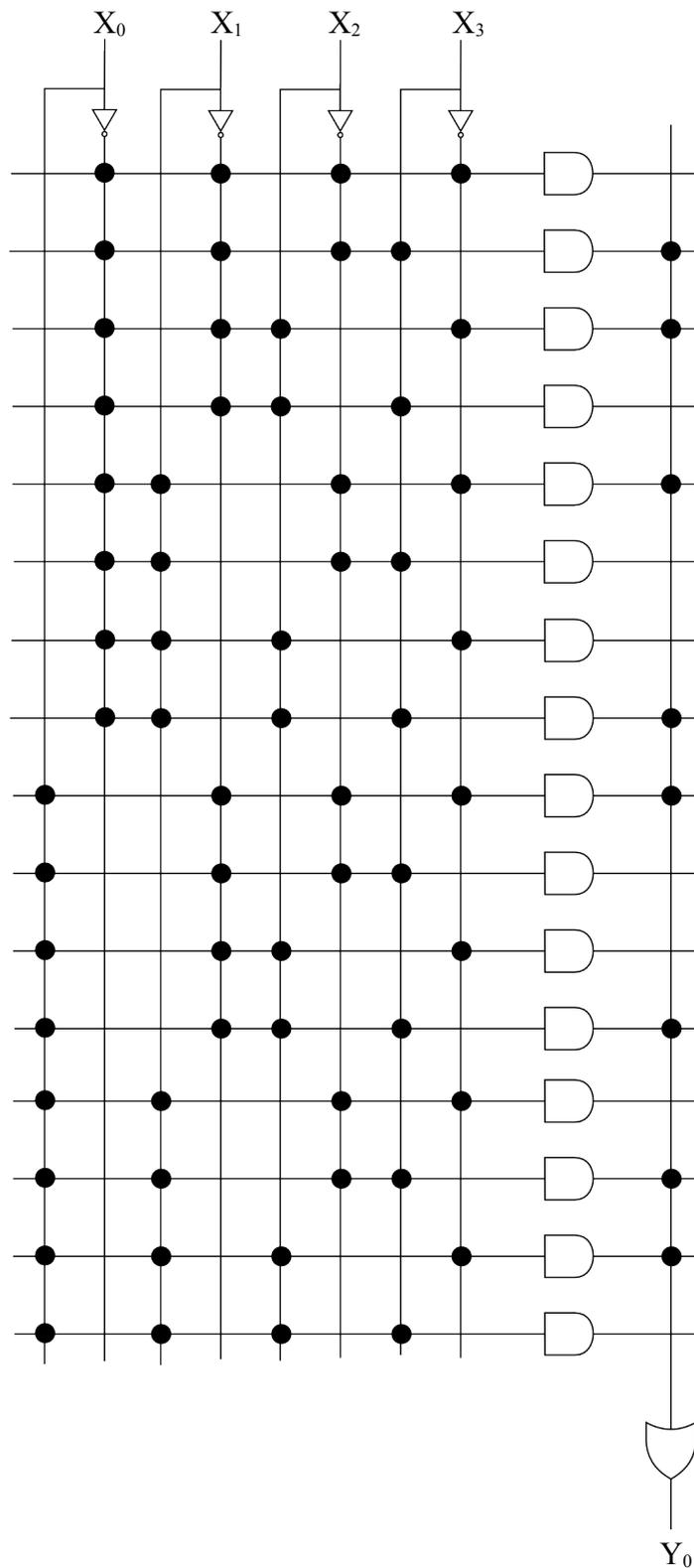


e) Si scriva la forma canonica disgiuntiva (SOP) per il bit meno significativo della funzione (2 punti)

Risultato:

$$Y_0 = \underline{X_0 X_1 X_2 X_3} + \underline{X_0 X_1 X_2 X_3}$$

f) Si realizzi tramite ROM l'espressione ottenuta al punto (e) (2 punti)
 (N.B.: è necessario specificare la matrice di AND e di OR)



g) Si realizzi tramite un MULTIPLEXER 16-a-1 il bit meno significativo della funzione (2 punti)
 (N.B.: si può assumere un modulo MUX che si comporta come un multiplexer 16-a-1;
 basta scriverne le entrate)

I bit di selezione corrisponderanno alle entrate X_0 X_1 X_2 X_3 .

I bit dei 16 canali di input saranno invece fissati ai valori della colonna Y_0 della tabella di verità:
 0110100110010110 (il primo bit può essere messo indifferentemente a 0 o a 1)